



EMBEDDED  
OPEN SOURCE  
SUMMIT



EMBEDDED  
LINUX  
CONFERENCE

# Tuning Real-Time Linux For ARM SoCs

**Slides:** <https://git.sr.ht/~bryanb/tuning-real-time-linux>

**Date:** 2024-04-18

**Bryan Brattlof**

**Pekka Varis**



TEXAS INSTRUMENTS

# About Us



Pekka Varis

- Senior Member Technical Staff



Bryan Brattlof

- Linux Baseport Software Developer

# About Us: TI Processors And Open Source



Decades of contribution and collaboration

Ingrained culture to give back to the community



## Upstream FIRST!

Focus on long term, sustainable and quality products



Upstream and open-source ecosystem in device architecture



Open  
Source

Upstream FIRST mentality!

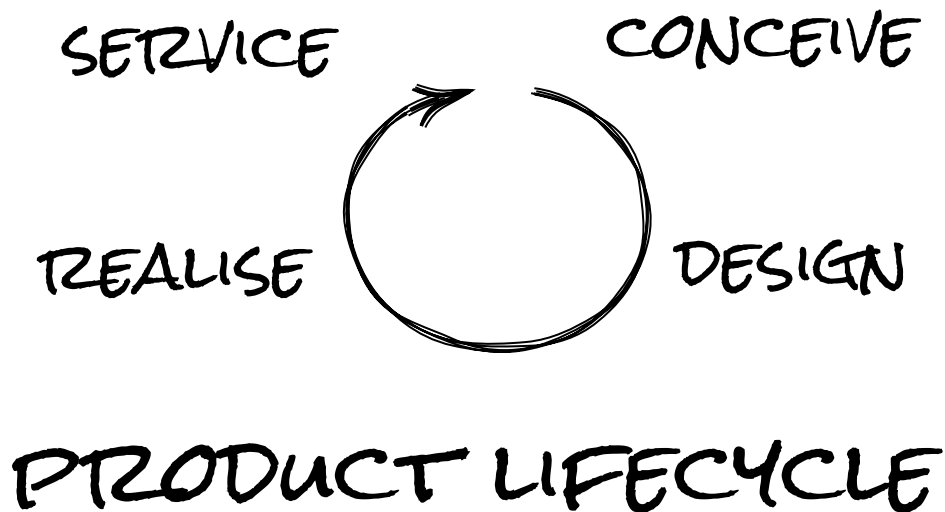


# I Can Get Into Enough Trouble Myself

This is a technology presentation, not  
product readiness or road-map commitments

These are my options and do not reflect  
Texas Instruments

# The Birth Of A New Project





**Wouldn't It Be Nice**

**You Need A TI SoC For That  
Real-Time Linux Application**

# Obligatory 'What Is Real-Time Linux'

- Maximize Preempt-ability

# Obligatory 'What Is Real-Time Linux'

- Maximize Preempt-ability
- Enable Scheduling From Anywhere



# Obligatory 'What Is Real-Time Linux'

- Maximize Preempt-ability
- Enable Scheduling From Anywhere
- Let The Most Important Task Run

# A Deterministic OS...

[HTTP://YouTube/Bx-Ujipcgcg](http://YouTube/Bx-Ujipcgcg)

Review

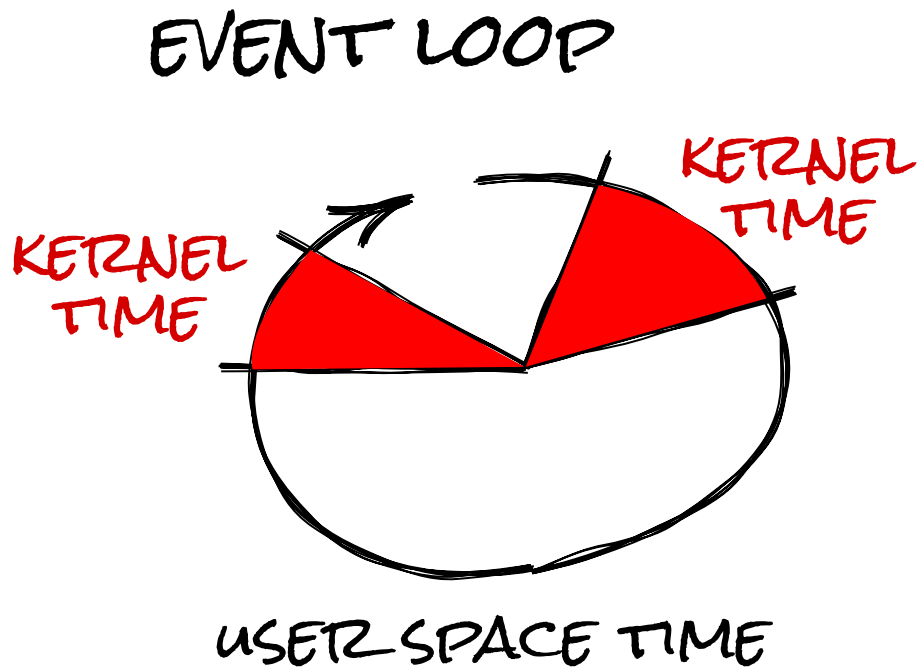
DOS!

vmware

 Embedded Linux  
Conference  
Europe

 OpenIoT Summit  
Europe

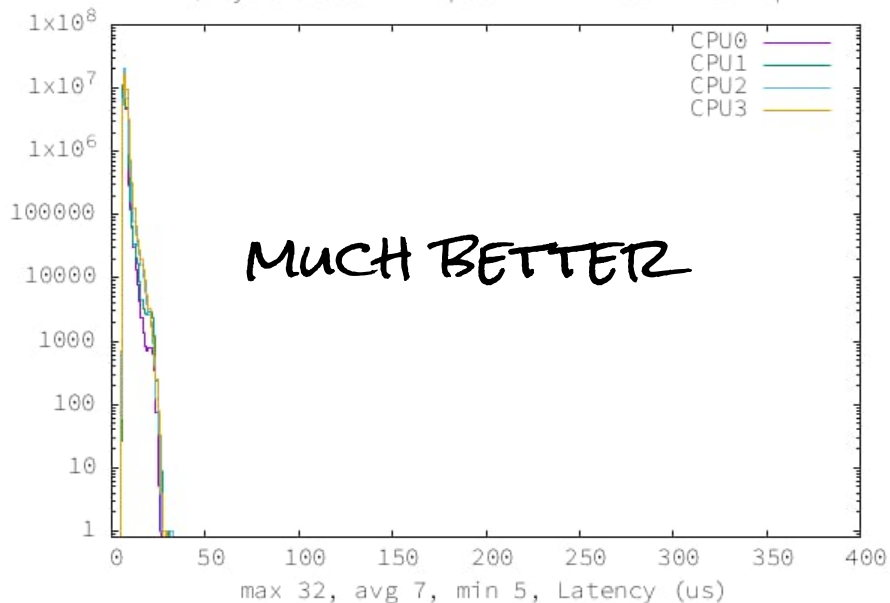
# We Can Never Get Rid Of Latency



# Limit Max Response Time

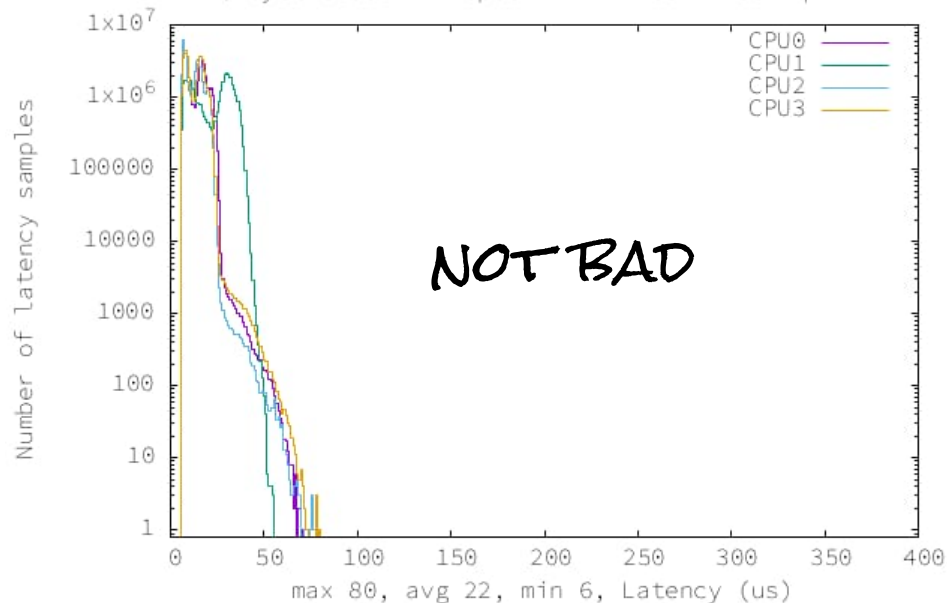
am62ax buildroot (32bit 1688MHz - 3376MT/s)

```
$ stress-ng --cpu-method=all -c4  
$ cyclictst -m -Sp80 -D2h -h400 -i200 -q
```



am62ax buildroot (32bit 1600MHz - 3200MT/s)


```
$ stress-ng --vm-method=zero-one --memrate 4  
$ cyclictst -m -Sp80 -D2h -h400 -i200 -q
```



# Bugs Are Everywhere

Gru is shown from the chest up, wearing his signature black hooded robe. He is pointing his right index finger upwards with a confident, slightly mischievous expression.

**We  
Find  
Bug**

Gru is shown from the chest up, wearing his signature black hooded robe. He is pointing his right index finger to his forehead with a thoughtful or perhaps slightly exasperated expression.

**We  
Fix  
Bug**

Gru is shown from the chest up, wearing his signature black hooded robe. He is leaning forward with his hands outstretched in front of him, looking directly at the viewer with a wide-eyed, intense expression.

**We  
Have  
Two  
Bugs**

Gru is shown from the chest up, wearing his signature black hooded robe. He is leaning forward, looking down at his hands which are positioned just above the clipboard, with a focused or perhaps slightly annoyed expression.

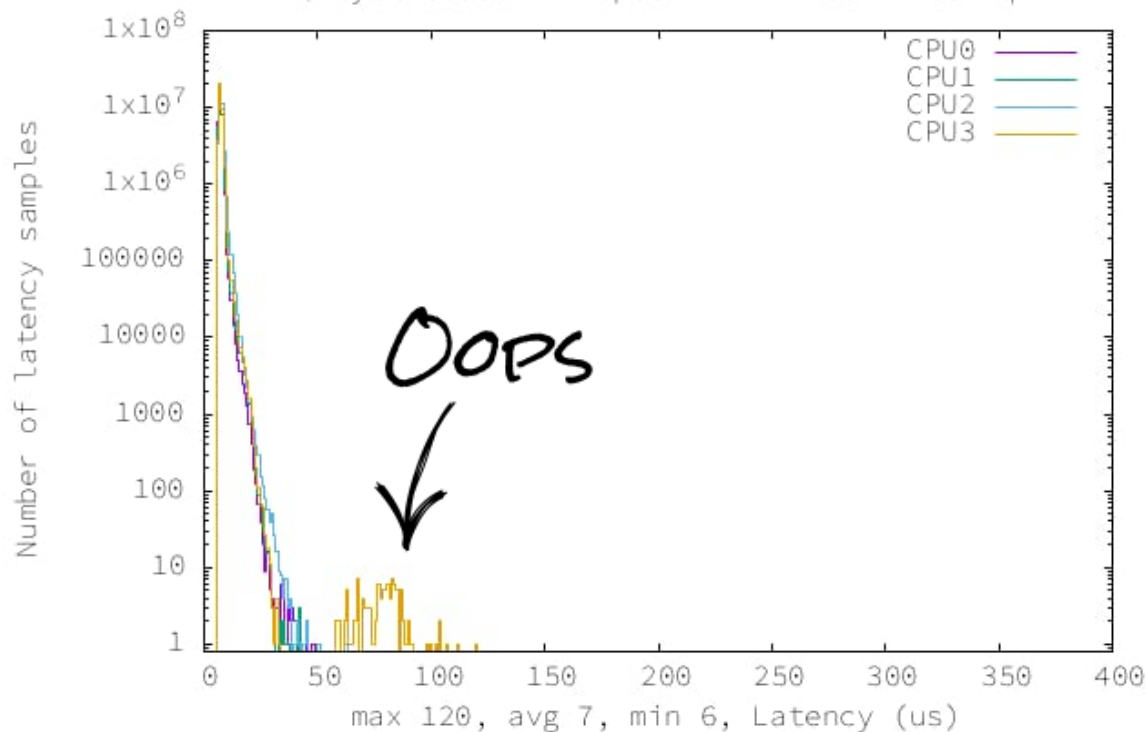
**We  
Have  
Three  
Bugs**

# Bugs Are Everywhere

am62ax yocto v09.00.00.009 (32bit 1688MHz - 3376MT/s)

```
$ stress-ng --cpu-method=all -c4
```

```
$ cyclictst -m -Sp80 -D2h -h400 -i200 -q
```



# plat-k3: sa2ul\_rng: Use mutex instead of spinlock for critical section #6599

[Code](#)

jforissier merged 1 commit into `OP-TEE:master` from `glneo:k3-sa2ul-rng-locking` on Jan 17



Conversation 5



Commits 1



Checks 7



Files changed 1

+3 -3

Changes from all commits ▾ File filter ▾ Conversations ▾ Jump to ▾ ⚙ ▾

Review in codespace

Review changes ▾

## ✓ plat-k3: sa2ul\_rng: Use mutex instead of spinlock for critical section

While spinlock are slightly more lightweight, they currently require that interrupts are disabled during the critical section. If this section is long enough it can have a negative affect on realtime sensitive tasks that require deterministic preemption.

As our RNG gathering can loop while waiting for new random numbers to become available we cannot know how long this section will take, so we should use a mutex. Do that here.

Signed-off-by: Andrew Davis <afd@ti.com>

Acked-by: Jerome Forissier <jerome.forissier@linaro.org>

Reviewed-by: Bryan Brattlof <bb@ti.com>

Acked-by: Jens Wiklander <jens.wiklander@linaro.org>

# Bugs Exists Outside Of Linux

glneo committed on Jan 16

commit 409055fdbadc3d4e088b30f3c3fef09c338c8f86



6



6

core/arch/arm/plat-k3/drivers/sa2ul\_rng.c



[https://github.com/OP-TEE/optee\\_os/pull/6599/commits/409055fdbadc3d](https://github.com/OP-TEE/optee_os/pull/6599/commits/409055fdbadc3d)

61

61

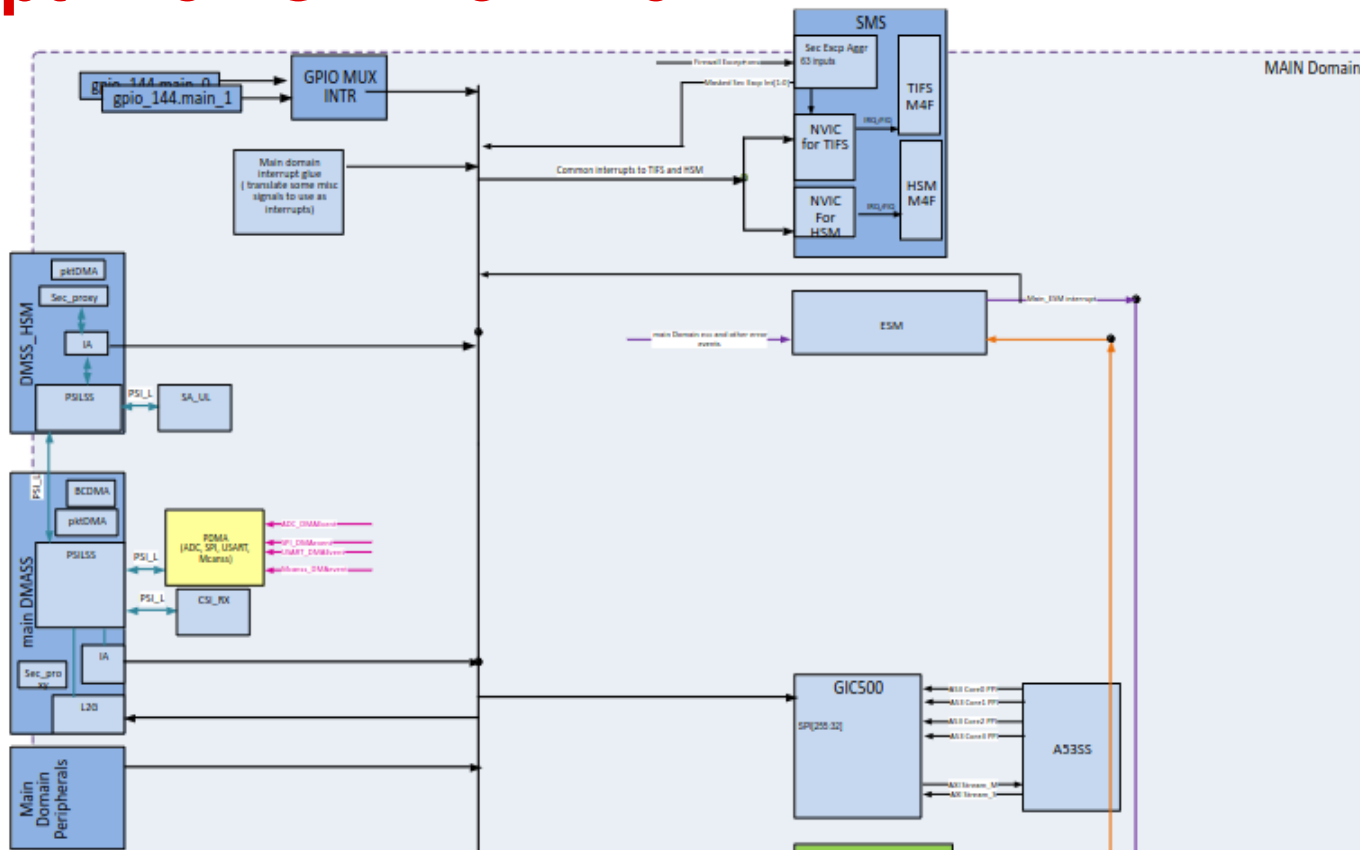


**Now What?**





# Interrupt Arch Of The AM62x



# The Easy

stress-ng  
hackbench

<https://github.com/ColinIanKing/stress-ng>

<https://git.kernel.org/pub/scm/utils/rt-tests/rt-tests.git>

# Timer Interrupts

```
$ stress-ng --cpu-method=all -c100 &  
$ rtpa timerlat hist
```

# Driver Interrupts (Ethernet)

```
$ iperf3 -c ${IP} -p ${PORT} -bidir -i 0
```

```
Connecting to host ${IP}, port ${PORT}
```

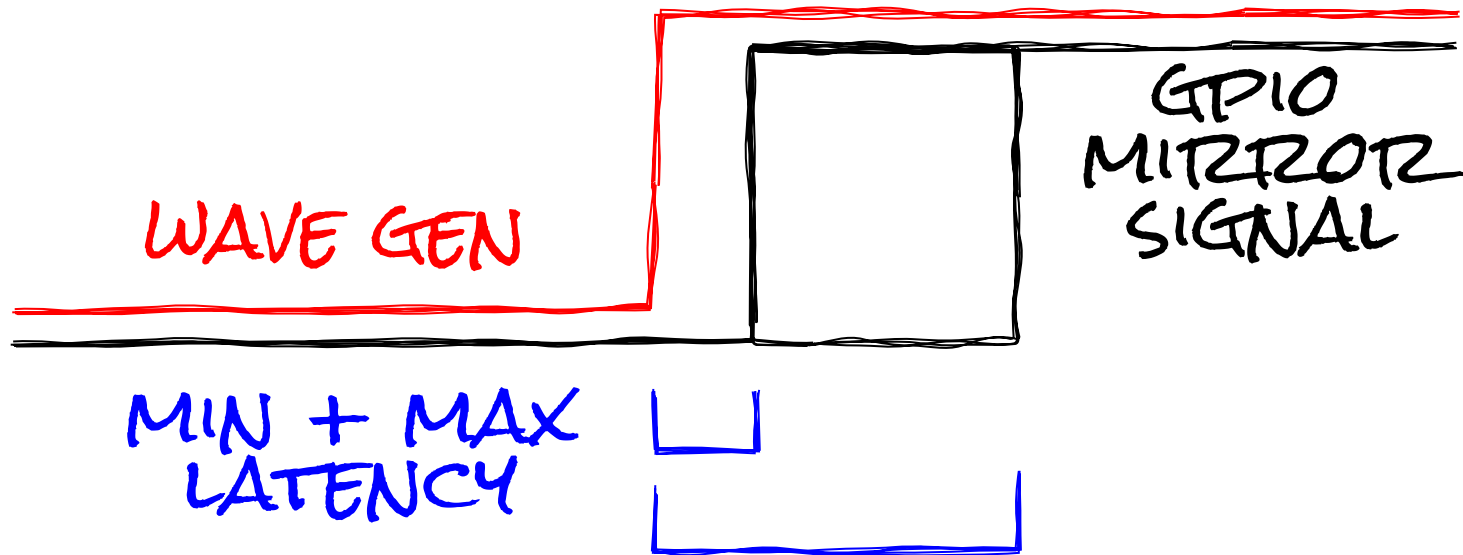
```
[ 5] local ${IP} port 38607 connected to ${IP} port ${PORT}
```

```
[ 7] local ${IP} port 38608 connected to ${IP} port ${PORT}
```

[ ID]	[Role]	Interval	Transfer	Bitrate	Retr	Cwnd
[ 5]	[TX-C]	0.00-10.00 sec	1.08 GBytes	931 Mbits/sec	0	1.04 MBytes
[ 7]	[RX-C]	0.00-10.00 sec	1.09 GBytes	933 Mbits/sec		

[ ID]	[Role]	Interval	Transfer	Bitrate	Retr	
[ 5]	[TX-C]	0.00-10.00 sec	1.08 GBytes	930 Mbits/sec	0	sender
[ 5]	[TX-C]	0.00-10.00 sec	1.08 GBytes	928 Mbits/sec		receiver
[ 7]	[RX-C]	0.00-10.00 sec	1.09 GBytes	934 Mbits/sec	0	sender
[ 7]	[RX-C]	0.00-10.00 sec	1.09 GBytes	933 Mbits/sec		receiver

# GPIO Interrupts



# Conclusions

Tips And Tricks

# Everything Is A Balance



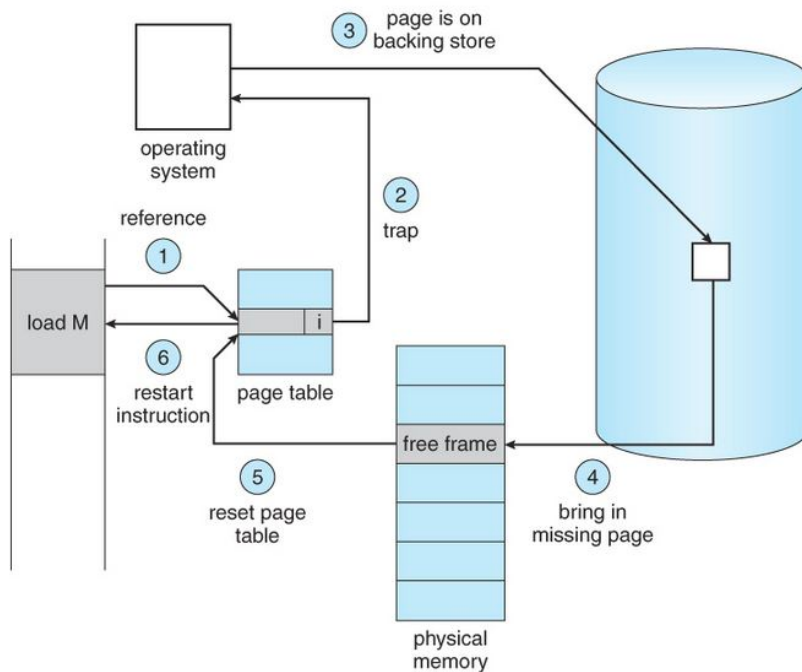
# Disable Unused Applications

```
# systemctl list-units
```



# Page Faults

Minimize the core kernel to  
minimize time spent on the  
slow path of the memory  
management subsystem



# Disable Everything

**+CONFIG\_KEYBOARD\_CROS\_EC=n**  
**+CONFIG\_I2C\_CROS\_EC\_TUNNEL=n**  
**+CONFIG\_QRTR=n**  
**+CONFIG\_SPI\_NXP\_FLEXPPI=n**  
**+CONFIG\_GPIO\_ALTERA=n**  
**+CONFIG\_GPIO\_DWAPB=n**  
**+CONFIG\_GPIO\_MB86S7X=n**  
**+CONFIG\_GPIO\_PL061=n**  
**+CONFIG\_GPIO\_XGENE=n**  
**+CONFIG\_POWER\_RESET\_XGENE=n**  
**+CONFIG\_ARM\_SP805\_WATCHDOG=n**  
**+CONFIG\_ARM\_SBSA\_WATCHDOG=n**  
**+CONFIG\_DW\_WATCHDOG=n**  
**+CONFIG\_MFD\_SEC\_CORE=n**  
**+CONFIG\_REGULATOR\_QCOM\_SPMI=n**

**+CONFIG\_CROS\_EC=n**  
**+CONFIG\_CROS\_EC\_I2C=n**  
**+CONFIG\_CROS\_EC\_SPI=n**  
**+CONFIG\_CROS\_EC\_CHARDEV=n**  
**+CONFIG\_SURFACE\_PLATFORMS=n**  
**+CONFIG\_COMMON\_CLK\_S2MPS11=n**  
**+CONFIG\_COMMON\_CLK\_VC5=n**  
**+CONFIG\_COMMON\_CLK\_BD718XX=n**  
**+CONFIG\_FSL\_ERRATUM\_A008585=n**  
**+CONFIG\_HISILICON\_ERRATUM\_161010101=n**  
**+CONFIG\_RPMMSG\_QCOM\_GLINK\_RPM=n**  
**+CONFIG\_SOUNDWIRE\_QCOM=n**  
**+CONFIG\_EXTCON\_USBC\_CROS\_EC=n**  
**+CONFIG\_QCOM\_SPMI\_ADC5=n**  
**+CONFIG\_IIO\_CROS\_EC\_SENSORS\_CORE=n**

# Mission Accomplished

**# Total: 108000000 107999931 107999901 107999793**  
**# Min Latencies: 00004 00004 00004 00004**  
**# Avg Latencies: 00005 00005 00005 00005**  
**# Max Latencies: 00033 00063 00046 00052**

# At What Cost

Every SoC vendor & app will have a  
different kernel configuration

# Q&A

- **Contact Information:**

- Bryan Brattlof <bb@ti.com>
- Pekka Varis <p-varis@ti.com>

- Also on IRC @ libera.chat #linux-ti

## Learn more about TI products

- <https://www.ti.com/linux>
- <https://www.ti.com/processors>
- <https://www.ti.com/edgeai>



### Why choose TI MCUs and processors?

#### ✓ Scalability

Our products offer scalable performance that can adapt and grow as the needs of your customers evolve.

#### ✓ Efficiency

We design products that extend battery life, maximize performance for every watt expended, and unlock the highest levels of system efficiency.

#### ✓ Affordability

We strive to make innovation accessible to all by creating cost-effective products that feature state-of-the-art technology and package designs.

#### ✓ Availability

Our investment in internal manufacturing capacity provides greater assurance of supply, supporting your growth for decades to come.